Optimization of a Mechanical Design Problem with the Modified Bacterial Foraging Algorithm

Efrén Mezura-Montes⋆, Edgar A. Portilla-Flores⋄, and
Betania Hernández-Ocaña⋆

⋆ Laboratorio Nacional de Informática Avanzada (LANIA) A.C.
Rébsamen 80, Centro, Xalapa, Veracruz, 91000, MEXICO.
⋄ Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC-IPN)
U. Adolfo López Mateos, México D.F., 07700, MEXICO.
emezura@lania.mx, aportilla@ipn.mx, betania_h_o@hotmail.com

Abstract. This paper presents the solution of a real-world constrained bi-objective mechanical design problem in presence of a dynamic constraint by using the Modified Bacterial Foraging Optimization Algorithm. This algorithm, originally designed to solve single-objective optimization problems, is adapted to include in its processes Pareto dominance as selection criterion, an external archive for elitism, and crowding distance as a diversity mechanism. The results obtained are compared with those provided by two evolutionary algorithms. The bacterial-based approach is able to provide a highly competitive performance and the type of solutions found are more suitable, based on the opinion of the expert mechanical designer.

Keywords: Engineering design, swarm intelligence, constraint-handling

1 Introduction

Nowadays, modern mechanical engineering designs are usually proposed as optimization problems [10, 14]. One common method to define such optimization problem, where the best possible combination of values to a set of parameters that describe the system are seek, requires a kinematic analysis as a first step. This kinematic analysis is used to fulfill positions and velocities of the mechanical system and allows it to be described by a set of parameters. In such analysis, performance functions and constraints to quantify the system behavior are considered. Once the set of parameters is defined, the designer can propose several potential solutions by himself. Otherwise, optimization algorithms can be used for that solution search [11].
Mathematical programming, such as the Newton method, is a valid option. In [8], an optimal kinematic design of a planar manipulator with four-bar mechanism was optimized with such method. Nonetheless, based on the inherent difficulty of the optimization problems derived from real-world mechanical systems, it may be difficult, in several cases, to solve them with mathematical programming methods. Therefore, nature-inspired meta-heuristic algorithms have become an

alternative option. Those algorithms can be roughly divided in two main classes [6]: (1) Evolutionary algorithms (EAs) and (2) swarm intelligence algorithms (SIAs).
Among SIAs, one of the still scarcely employed to solve mechanical design problems is the bacterial foraging optimization algorithm (BFOA), originally proposed to solve unconstrained single-objective optimization problems [12]. In BFOA, three steps based on the foraging behavior of E. Coli are emulated: chemotaxis (tumble and swimming), reproduction, and elimination-dispersal. In this paper, a simplified version proposed to solve single-objective constrained engineering design problems called Modified Bacterial Foraging Optimization Algorithm (MBFOA) [9] is further adapted to solve a bi-objective constrained mechanical design problem in presence of a dynamic constraint. Therefore, the contribution of the paper is two-fold: (1) to propose the first attempt to adapt MBFOA to solve multi-objective optimization problems and (2) to present a mechanism, based on a widely-used constraint-handling technique to deal with dynamic constraints in a real-world bi-objective optimization problem.

The paper is organized as follows: Section 2 describes the continuously variable transmission (CVT) system i.e., the problem tackled. Section 3 presents MBFOA. Thereafter, Section 4 introduces the modifications made to MBFOA so as to solve the CVT problem. Section 5 includes the obtained results and their corresponding discussion. Finally, Section 6 draws some conclusions and provides the future work.

2 CVT system

The mechanical design problem tackled in this paper is a crank-rocker-slider CVT system, which consists on a set of bar mechanisms carried out to perform simultaneously, amplifying or reducing the input speed. A chain system receives the motion from a input bar so as to keep a rotational movement in the same way the whole time. The CVT system is depicted in Figure 1 where the five design parameters are the following: r_1 is the ground bar, r_2 is the crank bar length, r_3 is the coupling bar length, r_4 is the rocker bar length and θ_1, is the angle between the reference bar and the horizontal axis. θ_2, θ_3, and θ_4 are calculated from the values of the five design parameters.

Considering that $x_1 = r_1$, $x_2 = r_2$, $x_3 = r_3$, $x_4 = r_4$, and $x_5 = \theta_1$ the constrained bi-objective optimization problem is defined as:

$$\text{Minimize } [f_1(\boldsymbol{x}), f_2(\boldsymbol{x})] \tag{1}$$

where f_1 represents the output of the system which aims to be maximized (it appears as a negative value because it is then considered for minimization):

$$f_1(x) = -(\theta_{4max} - \theta_{4min})^2 \tag{2}$$

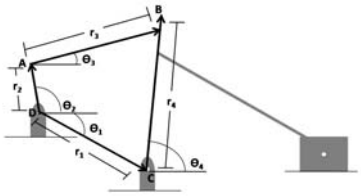

Fig. 1. CVT system

and f_2 minimizes the deviation of the transmission angle, i.e., it improves the quality of the performance of the CVT system:

$$f_2(x) = (\gamma_{max} - \frac{\pi}{2})^2 + (\gamma_{min} - \frac{\pi}{2})^2 \tag{3}$$

Furthermore, the inequality constraints related with the system dimensions, are the following:

$$g_1(\boldsymbol{x}) = x_2 + x_3 - x_1 - x_4 \leq 0 \tag{4}$$
$$g_2(\boldsymbol{x}) = x_1 - x_3 \leq 0 \tag{5}$$
$$g_3(\boldsymbol{x}) = x_4 - x_3 \leq 0 \tag{6}$$
$$g_4(\boldsymbol{x}) = x_1 - 0.5 \leq 0 \tag{7}$$
$$g_5(\boldsymbol{x}) = 0.05 - x_1 \leq 0 \tag{8}$$
$$g_6(\boldsymbol{x}) = x_2 - 0.5 \leq 0 \tag{9}$$
$$g_7(\boldsymbol{x}) = 0.05 - x_2 \leq 0 \tag{10}$$
$$g_8(\boldsymbol{x}) = x_3 - 0.5 \leq 0 \tag{11}$$
$$g_9(\boldsymbol{x}) = 0.05 - x_3 \leq 0 \tag{12}$$
$$g_{10}(\boldsymbol{x}) = x_4 - 0.05 \leq 0 \tag{13}$$
$$g_{11}(\boldsymbol{x}) = 0.05 - x_4 \leq 0 \tag{14}$$
$$g_{12}(\boldsymbol{x}) = x_5 - \frac{\pi}{4} \leq 0 \tag{15}$$
$$g_{13}(\boldsymbol{x}) = -\frac{\pi}{4} - x_5 \leq 0 \tag{16}$$

There is also one equality constraint defined as follows:

$$h_1(\boldsymbol{x}) = \pi - \theta_{4max} - \theta_{4min} = 0 \tag{17}$$

Finally, there is an inequality constraint related with the trajectory of the four bar mechanism where the transmission angle (determined by the values of θ_2 and θ_3) does not take a value below 45° in any single time.

$$g_{14}(\boldsymbol{x}, t) = \frac{\pi}{4} - \mu(x, t) \leq 0 \tag{18}$$

The limits for each design parameter are defined as follows: $0.05 \leq x_1, x_2, x_3, x_4 \leq 0.5$ and $-45^\circ \leq x_5 \leq 45^\circ$.

3 Modified Bacterial Foraging Optimization Algorithm (MBFOA)

The Bacterial Foraging Optimization (BFOA) is based on social and cooperative behaviors bacteria have when looking for regions with high nutrient levels [1]. The basis of the bacterial foraging behavior consists on that each bacterium tries to maximize its obtained energy per each unit of time spent on the foraging process while avoiding noxious substances. Moreover, bacteria can communicate among them. Due to space restrictions BFOA details are not included in this paper. However, they can be found in [12]

MBFOA was later proposed with the aim of adapting BFOA for solving constrained numerical single-objective optimization problems. This algorithm has a generational loop (G) where three inner processes are carried out: chemotaxis, reproduction and elimination-dispersal. MBFOA also adopted a parameter-free constraint-handling technique to bias the search to the feasible region of the search space to make it easier to use in engineering design problems [9]. The four main features of MBFOA are detailed as follows:

1. The constraint handling mechanism modifies the selection criteria used in the tumble-swim operator and in the sorting carried out in the reproduction process (comparisons originally based only on objective function values). In MBOFA a set of three rules proposed in [3] are employed. They are the following: (1) Between two feasible bacteria, the bacterium with the best objective function value is preferred. (2) Between a feasible bacterium and infeasible bacterium, the feasible one is preferred. (3) Between two infeasible bacteria, the one with the lowest sum of constraint violation is preferred. The sum of constraint violation is computed as: $\sum_{i=1}^{m} max(0, g_i(\boldsymbol{x}))$, where m is the number of constraints of the problem. Each equality constraint is converted into an inequality constraint: $\| h_i(x) \|$ -$\in \leq 0$, where $\in$ is the tolerance allowed (a very small value).
2. The chemotactic process consists on tumble-swim movements carried out by bacteria in the current swarm. The tumble movement (i.e., search direction chosen at random) is presented in Equation 19.

$$\phi(i) = \frac{\Delta(i)}{\sqrt{\Delta(i)^T \Delta(i)}} \tag{19}$$

where $\Delta_(i)$ is a randomly generated vector of size n with elements within the following interval: $[-1, 1]$. After that, each bacterium i modifies its positions by a swimming step as indicated in Equation 20.

$$\theta^i(j+1, G) = \theta^i(j, G) + C(i)\phi(i) \tag{20}$$

where $\theta^i(j+1,G)$ is the new position of bacterium i (new solution) at chemotactic step $j+1$, $\theta^i(j,G)$ is the current position of bacterium i at chemotactic step j.

Unlike considering them as user-defined parameters as in BFOA, in MBFOA the stepsize values in vector *C(i)* are calculated by considering the valid limits per each design variable k [9] as indicated in Equation 21.

$$C(i)_k = R * (\frac{\Delta \boldsymbol{x}_k}{\sqrt{n}}), k = 1, ..., n \tag{21}$$

where $\Delta \boldsymbol{x}_k$ is the difference between upper and lower limits for design parameter x_k: $U_k - L_k$, n is the number of design variables and R is a user-defined percentage of the value used by the bacteria as stepsize.
MBFOA implements an attractor movement so as to let each bacterium in the swarm to follow the bacterium located in the most promising region of the search space. The attractor movement is presented in Equation 22.

$$\theta^i(j+1, G) = \theta^i(j, G) + \beta(\theta^B(G) - \theta^i(j, G)) \tag{22}$$

where $\theta^i(j+1, G)$ is the new position of bacterium i, $\theta^i(j, G)$ is the current position of bacterium i, $\theta^B(G)$ is the current position of the best bacterium in the swarm so far at generation G, and β defines the closeness of the new position of bacterium i with respect to the position of the best bacterium $\theta^B(G)$. The attractor movement applies twice in a chemotactic loop, while in the remaining steps the tumble-swim movement is carried out. The aim is to promote a balance between exploration and exploitation in the search.

3. The reproduction process consists on sorting the swarm according to the rules of the constraint-handling technique. The first half survives while the second half is eliminated and the first half is duplicated to maintain a fixed swarm size.

4. The elimination-dispersal process eliminates only the worst bacterium, based on the three criteria defined in the constraint-handling technique, and a new randomly generated bacterium is inserted as its replacement.

4 Multi-Objective Modified Bacterial Foraging Optimization Algorithm (MOMBFOA)

MBFOA is modified in this section to solve the mechanical design problem presented in Section 2. following:

4.1 Evolutionary multi-objective optimization concepts

Based on the fact that the most suitable criterion to be added to a SIA so as to solve either two or three-objective optimization problem is *Pareto dominance*, it is considered in MOMBFOA. Pareto dominance is defined as follows: a vector of objectives $f = [f_1, \ldots, f_K]$ is said to Pareto dominate $f' = [f'_1, \ldots, f'_K]$ (denoted by $f \preceq f'$) if and only if f is partially less than f', i.e. $\forall i \in \{1, \ldots, K\},\ f_i \leq f'_i \wedge \exists i \in \{1, \ldots, K\} : f_i < f'_i$. The set of Pareto non-dominated solutions is

called the *Pareto optimal set*. The objective function values of those solutions contained in the Pareto optimal set constitute the *Pareto front* of the problem. The solution of a multi-objective problem can be defined as follows: If the feasible region of the search space is named as $\mathcal{F}$, the search algorithm will look for the Pareto optimal set: $\mathcal{P}^* := \{\boldsymbol{v} \in \mathcal{F} \mid \neg\exists\ \boldsymbol{s} \in \mathcal{F}\ \ f(\boldsymbol{s}) \preceq f(\boldsymbol{v})\}$. Considering that in this paper a real-world problem is solved, then $\mathcal{P}^*$ is unknown, a sub-optimal Pareto set, including sub-optimal trade-off solutions for the mechanical design problem, is the solution sought.

Another mechanism adopted from multi-objective optimization in this paper [4, 2], is the *crowding distance* [4]. *Crowding distance* values are higher for those more isolated solutions in the objective function space. Hence, with the goal to extend the Pareto front, bacteria with higher *crowding distance* values in the Archive will be preferred. Figure 2 details the computing of such distance.

Input data: Archive (A)
a=size(A) // store Archive size
For All (i ∈ A) **do**
  $A[i]_{dist}$=0 //initialize crowding distance for all bacteria to 0
End For
For All objective function f_j **do**
  A=sort(A,f_j) //Sort A in descending order according to f_j
  $A[1]_{dist} = A[a]_{dist} = \infty$ // ∞ value to bacteria in the extremes of the front for objective j
    For i=2 to a-1 **do** // for the remaining bacteria
      $A[i]_{dist} = A[i]_{dist} + \| \frac{f_j^{A[i-1]} - f_j^{A[i+1]}}{f_j^{max} - f_j^{min}} \|$ //compute distance value for objective j
    End For
End For

Fig. 2. Crowding distance pseudocode.

4.2 Modifications to MBFOA

The modifications are the following:

1. Selection criteria. Based on its similarity with the constraint-handling technique used in MBFOA, the feasibility rules described with Pareto dominance [5] is adopted in MOMBFOA. This criteria apply in the tumble-swim movements and in the sorting of the swarm. The set of criteria are defined as follows: (1) Between two feasible bacteria, the one which dominates the other is preferred. (2) Between one feasible bacterium and one infeasible bacterium, the feasible one is preferred. (3) Between two infeasible bacteria, the one with the lowest sum of constraint violation is preferred. If both bacteria are feasible and non-dominated each other, one is chosen randomly. To deal with the dynamic constraint, it was discretized in 911 constraints and each one was evaluated and its corresponding violation (if any) was added to the sum of constraint violation.
2. External Archive. Inspired in state-of-the-art algorithms for multi-objective optimization [16, 7] an external archive to only store feasible non-dominated

bacteria is employed in MOMBFOA. The Archive, empty at the beginning of the process, is updated at each MOMBFOA cycle by inserting a copy of those feasible bacteria from the swarm after the chemotactic process. Each time a set of bacteria enters the Archive, a non-dominance checking is carried out in the archive to keep feasible non-dominated bacteria only.

3. Attractor Operator. In MOMBFOA the leader will be chosen from the archive and it is the bacterium with the highest crowding distance value, i.e., the solution located in the most isolated region of the current Pareto front. The idea is promoting the sampling of more solutions in that region and extending the front. The new attractor operator is shown in Figure 23.

$$\theta^i(j+1,G) = \theta^i(j,G) + \beta(\theta^{BC_r}(G) - \theta^i(j,G)) \tag{23}$$

where $\theta^i(j+1,G)$ is the new position of the bacterium i, $\theta^i(j,G)$ is the current position of bacterium i, and $\theta^{BC_r}(G)$ is the current position of the bacterium with the best crowding distance in generation G taken from the external Archive. β is the stepsize of the attractor movement. If the Archive is empty, i.e., no feasible non-dominated solutions have been found, the leader is chosen from the current swarm and based on the criteria defined in Section 4.2, i.e., the bacterium with the lowest sum of constraint violation.

4. Reproduction. In MOMBFOA only the best bacterium in the swarm will reproduce with one clon which will replace the second worst bacterium, based on a sorting process computed by considering the criteria defined in Section 4.2. The complete MOMBFOA pseudocode is presented in Figure 3.

5 Experiments and results

Two experiments were designed to test MOMBFOA in the real-world mechanical problem stated in Section 2. Such results are compared with two well-known evolutionary multi-objective optimization algorithms: (1) NSGA-II [5] and (2) the differential evolution for mechanical design (DE) [13]. NSGA-II was chosen because it is the most popular EA to solve multi-objective problems. DE in [13] was precisely proposed to solve multi-objective mechanical design problems. Both algorithms chosen for comparison use crowding distance as diversity mechanism. DE uses an external archive for elitism purposes as in MOMBFOA.

The first experiment consists on a graphical comparison of the accumulated Pareto fronts obtained in a set of independent runs. The accumulated Pareto front is obtained by merging all non-dominated solutions for all independent runs and compute non-dominance checking. At the end, the set of non-dominated solutions from all the independent runs are showed in a single accumulated Pareto front. This way to visualize the results is useful in real-world multi-objective optimization problems where the Pareto optimal set is unknown.

The second experiment considers the usage of a binary performance metric to analyze the performance of the algorithms compared.

10 independent runs were carried out with the same set of parameters for MOMBFOA. The parameters used in all the experiments were obtained by a

```
Begin
Archive = ∅
Create a random initial swarm of bacteria θ^i(j,0) ∀i, i = 1,...,S_b
Evaluate f_k(θ^i(j,0)), g_m(θ^i(j,0)), ∀i, i = 1,...,S_b, ∀k, k = 1,...,K, ∀m, m = 1,...,M
For i=1 to S_b do
  If θ^i(j,0) is feasible
    Add θ^i(j,0) to the Archive by using non-dominance checking
  End If
End For
Compute the crowding distance for all solutions in the Archive
For G=1 to GMAX Do
  For i=1 to S_b Do
    For j=1 to N_c Do
      Perform the chemotaxis process(tumble-swim with Equations 19 and 20 and
      attractor operator with Equation 23 (or 22 if Archive=∅)) for bacteria θ^i(j,G) by
      considering the criteria defined in Section 4.2
    End For
  End For
  Perform the reproduction process by duplicating the best bacterium based on the
  criteria defined in Section 4.2 and deleting the second-to-last worst bacterium
  Perform the elimination-dispersal process by eliminating the worst bacterium
  θ^w(j,G) in the current swarm by considering the criteria defined in Section 4.2
  Add the feasible non-dominated bacteria to the Archive by using non-dominance checking
  Compute the crowding distance for all solutions in the Archive
End For
End
```

Fig. 3. MOMBFOA pseudocode. Input parameters are number of bacteria S_b, chemotactic loop limit N_c, scaling factor β, percentage of initial stepsize R and number of generations $GMAX$.

trial-and-error process based on preliminary sets or runs. They are the following: $S_b = 200$, $GMAX = 125$, $N_c = 20$, $F = 1.7$ and $R = 1.8\text{E-}3$. 500, 000 evaluations are performed in each independent run. The tumble-swim operator works in all chemotactic steps for each bacterium with the exception of steps 15 and 20, where the attraction operator is used. The results obtained by NSGA-II and DE were obtained from 10 independent runs with 500,000 evaluations as well.

The accumulated Pareto fronts for the first experiment are presented in Figure 4, where it can be clearly observed that the accumulated front obtained by MOMBFOA dominates those obtained by NSGA-II and DE. However, DE has a larger front than that of MOMBFOA. However, from the mechanical engineer's point of view, the lower right part of the objective space has more suitable solutions and precisely that region was found by MOMBFOA. Another significant finding was that all compared algorithms could find the feasible region of the search space despite the presence of the dynamic constraint showed in Section 2.

The two-set coverage metric [15] was used to compare each one of the single Pareto fronts obtained by each algorithm and the statistical summary of such results, besides the statistical tests to verify significance of differences, are showed in Table 1.

The results in Table 1 suggests that MOMBFOA and DE outperformed NSGA-II. On the other hand, MOMBFOA and DE provided a similar behavior regarding the dominance of solutions between them.

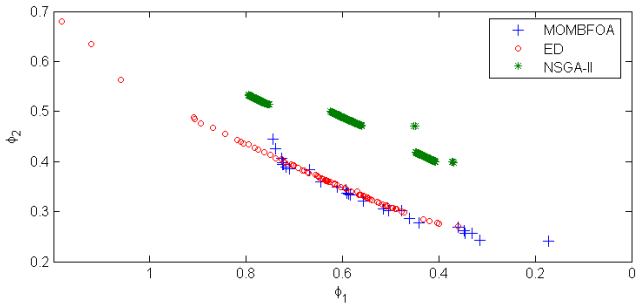

Fig. 4.

Algorithms	Best	Mean	Median	St.Dev	Worse	U Mann of Whitney
MOMBFOA-NSGA-II	**1**	1	1	**0**	1	
NSGA-II-MOMBFOA	0	0	0	0	0	$\surd$
DE-NSGA-II	**1**	1	1	**0**	1	
NSGA-II-DE	0	0	0	0	0	$\surd$
MOMBFOA-DE	0.9259	0.3366	0.3537	0.2571	0	
DE-MOMBFOA	1	0.3283	0.25	0.3266	0	$=$

Table 1. Summary of results for the two-set coverage metric. A higher value is preferred. '$\surd$' represents a significant difference and '$=$' indicates no significant differences. Mann-Whitney U non-parametric test was used.

6 Conclusions and future work

In this work the MBFOA algorithm was implemented in order to carry out the mechanical design of a bi-objective optimization problem of a four-bar mechanism. Such design considered two objective functions and structural constraints, where one of them is a dynamic constraint. The results obtained by MOMBFOA were compared with two algorithms: NSGA-II and DE, in two experiments, one with a visual comparison of the accumulated Pareto fronts and another with the usage of the two-set coverage metric and the Mann Whitney U test to verify significance of results. Two interesting conclusions are established from the discussed results: (1) MOMBFOA was able to find a region of the objective function space while the two other algorithms could not reach it and such region reflects a more suitable design of the mechanism, and (2) the performance metric was unable to measure such user-preference. The future work consists on using MOMBFOA to resolve other multi-objective optimization problems. Moreover, a performance metric which can be able to measure user-preferences in mechanical design must be defined.

Acknowledgments

The first and third authors acknowledge support from CONACyT through project No. 79809, and through a scholarship to pursue graduate studies at LANIA, re-

spectively. The second author acknowledges support from IPN's programs COFAA and EDI, and from SIP through project numbers 20101360 and 20111027.

References

1. H. Bremermann. Chemotaxis and optimization. *J. Franklin Inst.*, 297:397–404, 1974.
2. C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, June 2002.
3. K. Deb. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2/4):311–338, 2000.
4. K. Deb. *Multi-Objetive Optimization using Evolutionary Algorithms*. John Wiley Sons, LTD, 2002.
5. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.
6. A. P. Engelbrecht. *Computational Intelligence. An Introduction*. John Wiley & Sons, 2nd edition, 2007.
7. J. D. Knowles and D. W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
8. V. Mermetas. Optimal design of manipulator with four-bar mechanism. *Mechanism and Machine Theory*, 1(39):545–554, April 2004.
9. E. Mezura-Montes and B. Hernández-Ocaña. Modified bacterial foraging optimization for engineering design. In C. H. Dagli and et al., editors, *Proceedings of the Artificial Neural Networks in Enginnering Conference (ANNIE'2009)*, pages 357–364, St. Louis, MO, USA, November 2009. ASME Press.
10. N. Nariman-Zadeh, M. Felezi, A. Jamali, and M. Ganji. Pareto optimal synthesisof four-bar for path generation. *Mechanism and Machine Theory*, 1(44):180–191, March 2009.
11. R. Norton. *Diseño de Maquinaria, una introducción a la síntesis y al análisis de mecanismos y máquinas*. McGraw Hill, Mxico, 1997.
12. K. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3):52–67, 2002.
13. E. A. Portilla-Flores, E. Mezura-Montes, J. Álvarez Gallegos, C. A. C. Coello, C. A. Cruz-Villar, and M. G. Villareal-Cervantes. Parametric reconfiguration improvement in non-iterative concurrent mechatronic design using an evolutionary-based approach. *Engineering Optimization of Artificial Intelligence*, 24(5):757–771, 2011.
14. R. Radovan and R. Stevan. On the optimum synthesis of a four-bar linkage usingdifferential evolution and method of variable controlled deviations. *Mechanism and Machine Theory*, 1(44):235–246, March 2009.
15. E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. evolutionary computation. *Evolutionary Computation*, 8(2):173–195, 2000.
16. E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.